

# HelloBot REST API Guide

Build by "Mojunseo"

---

## Table of Contents

개요

HTTP 동사

HTTP 상태 코드

하이퍼미디어

리소스

ChatBot API

사용자 정보가져오기

HTTP request

Example response

Response fields

Links

시나리오 만들기

HTTP request

Example response

Response fields

Links

시나리오 시작하기

HTTP request

Example response

Response fields

Links

사용자의 빠른입력(버튼)에 따른 챗봇 응답

HTTP request

Example response

Request fields

Response fields

Links

사용자의 자유입력에 따른 챗봇 응답

HTTP request

Example response

Request fields

Response fields

Links

단위/통합테스트

테스트 케이스

---

## 개요

### HTTP 동사

본 REST API에서 사용하는 HTTP 동사(verbs)는 가능한한 표준 HTTP와 REST 규약을 따릅니다.

동사	정의
----	----

동사	정의
GET	리소스를 가져올 때 사용
POST	새 리소스를 만들 때 사용

## HTTP 상태 코드

본 REST API에서 사용하는 HTTP 상태 코드는 가능한한 표준 HTTP와 REST 규약을 따릅니다.

상태 코드	용례
200 OK	요청을 성공적으로 처리함
201 Created	새 리소스를 성공적으로 생성함. 응답의 Location 헤더에 해당 리소스의 URI가 담겨있다.
400 Bad Request	잘못된 요청을 보낸 경우. 응답 본문에 더 오류에 대한 정보가 담겨있다.
404 Not Found	요청한 리소스가 없음.

## 하이퍼미디어

본 REST API는 하이퍼미디어와 사용하며 응답에 담겨있는 리소스는 다른 리소스에 대한 링크를 가지고 있다.

응답은 [Hypertext Application from resource to resource. Language \(HAL\)](#) 형식을 따른다.

링크는 `\_links`라는 키로 제공한다. 본 API의 사용자(클라이언트)는 URI를 직접 생성하지 않아야 하며, 리소스에서 제공하는 링크를 사용해야 한다.

## 리소스

### ChatBot API

챗봇API는 사용자의 입력에 따라 정해진 시나리오를 대답해주는 API입니다.

### 사용자 정보가져오기

Get 요청을 사용해서 생성된 사용자 정보를 가져올 수 있습니다.

반환 된 Token을 이용하여 리소스에 접근할 수 있습니다.

#### HTTP request

```
GET /api/chat HTTP/1.1
Host: localhost:8080
```

#### Example response

HTTP/1.1 201 Created  
Pragma: no-cache  
Location: http://localhost:8080/api/chat  
X-XSS-Protection: 1; mode=block  
Access-Control-Allow-Methods: POST, GET, OPTIONS, DELETE, PUT  
Expires: 0  
X-Content-Type-Options: nosniff  
Content-Type: application/hal+json;charset=UTF-8  
Access-Control-Allow-Headers: \*  
Access-Control-Max-Age: 3600  
Content-Length: 524  
Access-Control-Allow-Origin: \*  
Cache-Control: no-cache, no-store, max-age=0, must-revalidate

```
{
  "content" : {
    "id" : "admin",
    "name" : "제이마인드",
    "birth" : "1992-02-28",
    "sex" : "Male",
    "keyword" : "책임감있고 신중한",
    "ChatBotToken" : "87f11ac3-23a0-43c6-bc33-7162ae0d2f2f"
  },
  "_links" : {
    "self" : {
      "href" : "http://localhost:8080/api/chat"
    },
    "시나리오 만들기 링크" : {
      "href" : "http://localhost:8080/api/chat/makeScenario"
    },
    "profile" : {
      "href" : "/docs/index.html#resource-getAccount"
    }
  }
}
```

## Response fields

Path	Type	Description
content	Object	Account Info
content.id	String	User ID
content.name	String	User Name
content.birth	String	User Birth
content.sex	String	User Sex
content.keyword	String	User Keyword
content.ChatBotToken	String	User ChatBot-Token

## Links

Relation	Description
self	현재 링크
시나리오 만들기 링크	시나리오 만들기 링크
profile	도큐먼트 링크

## 시나리오 만들기

`Get` 요청을 사용해서 미리 정해진 시나리오를 생성할 수 있습니다.

### HTTP request

```
GET /api/chat/makeScenario HTTP/1.1
Authorization: Bearer 981bcb7c-a6ee-4296-9b4f-6f03711408eb
Host: localhost:8080
```

### Example response

```
HTTP/1.1 201 Created
Pragma: no-cache
Location: http://localhost:8080/api/chat
X-XSS-Protection: 1; mode=block
Access-Control-Allow-Methods: POST, GET, OPTIONS, DELETE, PUT
Expires: 0
X-Content-Type-Options: nosniff
Content-Type: application/hal+json;charset=UTF-8
Access-Control-Allow-Headers: *
Access-Control-Max-Age: 3600
Access-Control-Allow-Origin: *
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Content-Length: 2964

{
  "account" : {
    "id" : "adminTDD",
    "name" : "제이마인드",
    "birth" : "1992-02-28",
    "sex" : "Male",
    "keyword" : "책임감있고 신중한",
    "ChatBotToken" : null
  },
  "scenario" : [ {
    "id" : 1,
    "type" : [ "MULTIPART", "TEXT" ],
    "responseContent" : "안녕!나는 개발남이야ㅎㅎ만나게 되어서 반가워\n나와 함께 개발자가 되어볼래?",
    "imageFilePath" : "https://jayass3cloud.s3.ap-northeast-2.amazonaws.com/IMG_8456.jpg",
    "responseAnswer" : "응!해볼래"
  }, {
    "id" : 2,
    "type" : [ "TEXT" ],
    "responseContent" : "$NAME$는 보아하니 책임감있고 신중해보여!\n그래서 나는 보이지는 않지만 데이터를 효율적으로 관리하고",
    "imageFilePath" : "EMPTY",
    "responseAnswer" : "응!좋아"
  }, {
    "id" : 3,
    "type" : [ "TEXT" ],
    "responseContent" : "냥이는 $NAME$한테 Spring Boot Framework를 이용하는 Java개발자가 되는게 좋아보이는데\n혹시 Java",
    "imageFilePath" : "EMPTY",
    "responseAnswer" : "응!문서를 참고해서 간단한 서비스개발가능해"
  }, {
    "id" : 4,
    "type" : [ "MULTIPART", "TEXT" ],
    "responseContent" : "$IMAGE$\n오호~대단한데? 그럼 RESTful API를 개발해보자!\nRESTful API가 뭔지알아?",
    "imageFilePath" : "https://lh3.googleusercontent.com/proxy/9B4_NmGyPZCtJ0taWAKbcnuqs1VtDPBozlCwm9CBogaMUFr",
    "responseAnswer" : "아니 잘 모르겠어!"
  }, {
    "id" : 5,
    "type" : [ "TEXT" ],
    "responseContent" : "현재 REST API로 불리는 대부분의 API가 실제로는 로이 필딩이 정의한 REST를 따르고 있지 않고있어.",
    "imageFilePath" : "EMPTY",
    "responseAnswer" : "응!다 했어"
  }, {
    "id" : 6,
    "type" : [ "TEXT" ],
    "responseContent" : "$NAME$은 백엔드 개발자로 폭풍 성장중입니다..",
    "imageFilePath" : "EMPTY",
    "responseAnswer" : "응!대박"
```

```
    "responseAnswer" : "..."  
  } ],  
  "_links" : {  
    "self" : {  
      "href" : "http://localhost:8080/api/chat/makeScenario"  
    },  
    "시나리오 시작 링크" : {  
      "href" : "http://localhost:8080/api/chat/1"  
    },  
    "profile" : {  
      "href" : "/docs/index.html#resource-makeScenario"  
    }  
  }  
}
```

## Response fields

Path	Type	Description
account	Object	Account Info
account.id	String	User ID
account.name	String	User Name
account.birth	String	User Birth
account.sex	String	User Sex
account.keyword	String	User Keyword
account.ChatBotToken	Null	User ChatBot-Token
scenario	Array	Scenario Data
scenario[0].id	Number	Scenario Depth
scenario[0].type	Array	Bot Response Type
scenario[0].responseContent	String	Bot Response TEXT
scenario[0].imageFilePath	String	Bot Response of Image Path
scenario[0].responseAnswer	String	Bot Response of Button

## Links

Relation	Description
self	현재 링크
시나리오 시작 링크	시나리오 시작 링크
profile	도큐먼트 링크

## 시나리오 시작하기

Post 요청을 사용해서 미리 정해진 시나리오를 생성할 수 있습니다.  
시나리오 시작은 Dto를 요구하지 않습니다.

## HTTP request

```
POST /api/chat/1 HTTP/1.1
Host: localhost:8080
Authorization: Bearer a83980f1-0c77-465d-b677-82eefb71b3f0
```

## Example response

```
HTTP/1.1 201 Created
Pragma: no-cache
Location: http://localhost:8080/api/chat
X-XSS-Protection: 1; mode=block
Access-Control-Allow-Methods: POST, GET, OPTIONS, DELETE, PUT
Expires: 0
X-Content-Type-Options: nosniff
Content-Type: application/hal+json;charset=UTF-8
Access-Control-Allow-Headers: *
Content-Length: 830
Access-Control-Max-Age: 3600
Access-Control-Allow-Origin: *
Cache-Control: no-cache, no-store, max-age=0, must-revalidate

{
  "account" : {
    "id" : "adminTDD",
    "name" : "제이마인드",
    "birth" : "1992-02-28",
    "sex" : "Male",
    "keyword" : "책임감있고 신중한",
    "ChatBotToken" : null
  },
  "chatResponse" : {
    "id" : 1,
    "type" : [ "MULTIPART", "TEXT" ],
    "responseContent" : "안녕!나는 개발냥이야ㅎㅎ만나게 되어서 반가워\n나와 함께 개발자가 되어볼래?",
    "imageFilePath" : "https://jayass3cloud.s3.ap-northeast-2.amazonaws.com/IMG_8456.jpg",
    "responseAnswer" : "응!해볼래"
  },
  "_links" : {
    "self" : {
      "href" : "http://localhost:8080/api/chat/1"
    },
    "다음 단계 링크" : {
      "href" : "http://localhost:8080/api/chat/2"
    },
    "profile" : {
      "href" : "/docs/index.html#resource-chatProcessStart"
    }
  }
}
```

## Response fields

Path	Type	Description
account	Object	Account Info
account.id	String	User ID
account.name	String	User Name
account.birth	String	User Birth
account.sex	String	User Sex

Path	Type	Description
account.keyword	String	User Keyword
account.ChatBotToken	Null	User ChatBot-Token
chatResponse	Object	ChatBot Response Data
chatResponse.id	Number	ChatBot Response Depth
chatResponse.type	Array	ChatBot Response Type
chatResponse.responseContent	String	ChatBot Response TEXT. divide by \n
chatResponse.imageFilePath	String	ChatBot Response of Image Path
chatResponse.responseAnswer	String	ChatBot Response of Button

## Links

Relation	Description
self	현재 링크
다음 단계 링크	다음 단계 링크
profile	도큐먼트 링크

## 사용자의 빠른입력(버튼)에 따른 챗봇 응답

Post 요청을 사용해서 미리 정해진 시나리오를 진행합니다.

빠른입력(버튼)을 이용하면 미리 정해둔 값에만 응답합니다.

## HTTP request

```
POST /api/chat/2 HTTP/1.1
Content-Length: 60
Content-Type: application/json; charset=UTF-8
Host: localhost:8080
Authorization: Bearer a83980f1-0c77-465d-b677-82eefb71b3f0

{
  "type" : "BUTTON",
  "userAnswer" : "응!해볼래"
}
```

## Example response

```
HTTP/1.1 201 Created
Pragma: no-cache
Content-Length: 938
Location: http://localhost:8080/api/chat
X-XSS-Protection: 1; mode=block
Access-Control-Allow-Methods: POST, GET, OPTIONS, DELETE, PUT
Expires: 0
X-Content-Type-Options: nosniff
Content-Type: application/hal+json;charset=UTF-8
Access-Control-Allow-Headers: *
Access-Control-Max-Age: 3600
Access-Control-Allow-Origin: *
Cache-Control: no-cache, no-store, max-age=0, must-revalidate

{
  "account" : {
    "id" : "adminTDD",
    "name" : "제이마인드",
    "birth" : "1992-02-28",
    "sex" : "Male",
    "keyword" : "책임감있고 신중한",
    "ChatBotToken" : null
  },
  "chatResponse" : {
    "id" : 2,
    "type" : [ "TEXT" ],
    "responseContent" : "$NAME$는 보아하니 책임감있고 신중해보여!\n그래서 나는 보이지는 않지만 데이터를 효율적으로 관리하고",
    "imageFilePath" : "EMPTY",
    "responseAnswer" : "응! 좋아"
  },
  "_links" : {
    "self" : {
      "href" : "http://localhost:8080/api/chat/2"
    },
    "이전 단계 링크" : {
      "href" : "http://localhost:8080/api/chat/1"
    },
    "다음 단계 링크" : {
      "href" : "http://localhost:8080/api/chat/3"
    },
    "profile" : {
      "href" : "/docs/index.html#resource-chatProcess"
    }
  }
}
```

## Request fields

Path	Type	Description
type	String	Type of Message
userAnswer	String	User Answer

## Response fields

Path	Type	Description
account	Object	Account Info
account.id	String	User ID
account.name	String	User Name



Path	Type	Description
account.birth	String	User Birth
account.sex	String	User Sex
account.keyword	String	User Keyword
account.ChatBotToken	Null	User ChatBot-Token
chatResponse	Object	ChatBot Response Data
chatResponse.id	Number	ChatBot Response Depth
chatResponse.type	Array	ChatBot Response Type
chatResponse.responseContent	String	ChatBot Response TEXT. divide by \n
chatResponse.imageFilePath	String	ChatBot Response of Image Path
chatResponse.responseAnswer	String	ChatBot Response of Button

## Links

Relation	Description
self	현재 링크
이전 단계 링크	이전 단계 링크
다음 단계 링크	다음 단계 링크
profile	도큐먼트 링크

## 사용자의 자유입력에 따른 챗봇 응답

Post 요청을 사용해서 미리 정해진 시나리오를 진행합니다.  
자유입력 이용하면 미리 정해둔 값과 유사한 경우에 응답합니다.

## HTTP request

```
POST /api/chat/2 HTTP/1.1
Content-Length: 60
Content-Type: application/json; charset=UTF-8
Host: localhost:8080
Authorization: Bearer a83980f1-0c77-465d-b677-82eefb71b3f0

{
  "type" : "BUTTON",
  "userAnswer" : "응!해볼래"
}
```

## Example response

```
HTTP/1.1 201 Created
Pragma: no-cache
Content-Length: 938
Location: http://localhost:8080/api/chat
X-XSS-Protection: 1; mode=block
Access-Control-Allow-Methods: POST, GET, OPTIONS, DELETE, PUT
Expires: 0
X-Content-Type-Options: nosniff
Content-Type: application/hal+json;charset=UTF-8
Access-Control-Allow-Headers: *
Access-Control-Max-Age: 3600
Access-Control-Allow-Origin: *
Cache-Control: no-cache, no-store, max-age=0, must-revalidate

{
  "account" : {
    "id" : "adminTDD",
    "name" : "제이마인드",
    "birth" : "1992-02-28",
    "sex" : "Male",
    "keyword" : "책임감있고 신중한",
    "ChatBotToken" : null
  },
  "chatResponse" : {
    "id" : 2,
    "type" : [ "TEXT" ],
    "responseContent" : "$NAME$는 보아하니 책임감있고 신중해보여!\n그래서 나는 보이지는 않지만 데이터를 효율적으로 관리하고",
    "imageFilePath" : "EMPTY",
    "responseAnswer" : "응! 좋아"
  },
  "_links" : {
    "self" : {
      "href" : "http://localhost:8080/api/chat/2"
    },
    "이전 단계 링크" : {
      "href" : "http://localhost:8080/api/chat/1"
    },
    "다음 단계 링크" : {
      "href" : "http://localhost:8080/api/chat/3"
    },
    "profile" : {
      "href" : "/docs/index.html#resource-chatProcess"
    }
  }
}
```

## Request fields

Path	Type	Description
type	String	Type of Message
userAnswer	String	User Answer

## Response fields

Path	Type	Description
account	Object	Account Info
account.id	String	User ID
account.name	String	User Name

Path	Type	Description
account.birth	String	User Birth
account.sex	String	User Sex
account.keyword	String	User Keyword
account.ChatBotToken	Null	User ChatBot-Token
chatResponse	Object	ChatBot Response Data
chatResponse.id	Number	ChatBot Response Depth
chatResponse.type	Array	ChatBot Response Type
chatResponse.responseContent	String	ChatBot Response TEXT. divide by \n
chatResponse.imageFilePath	String	ChatBot Response of Image Path
chatResponse.responseAnswer	String	ChatBot Response of Button

#### Links

Relation	Description
self	현재 링크
이전 단계 링크	이전 단계 링크
다음 단계 링크	다음 단계 링크
profile	도큐먼트 링크

## 단위/통합테스트

여러 경우에 대비하여 단위 테스트와 통합테스트를 진행했습니다.

### 테스트 케이스

1. 사전에 생성된 Account 계정 정보를 가져오는 API테스트
2. 잘못된 토큰을 가지고 허용되지않은 자원에 접근할 경우의 API테스트
3. 미리 작성된 시나리오를 생성하는 API테스트
4. 채팅 시작시, 응답을 가져오는 API테스트
5. 각 대화 Depth(단계)에 따라, 사용자가 빠른선택지를 이용했을 때 응답하는 API테스트
6. 각 대화 Depth(단계)에 따라, 사용자가 자유입력형식을 이용했을 때 응답하는 API테스트
7. 각 대화 Depth(단계)에 따라, 사용자가 '잘못된' 빠른선택지를 이용했을 때 응답하는 API테스트
8. 각 대화 Depth(단계)에 따라, 사용자가 이용한 자유입력이 정해진 응답과 방향이 다를 경우 API테스트
9. 각 대화 Depth(단계)에 따른 프로세스 진행시, 온전하지 못한 Dto를 전송할 경우의 API테스트

